CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Predicting the performance of job applicants in coding tests

Bachelor of Science Thesis in Software Engineering and Management

RACHELE MELLO

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Predicting the performance of job applicants in coding tests

RACHELE MELLO

Supervisor: JAN-PHILIPP STEGHÖFER
Examiner: ERIC KNAUSS

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Predicting the performance of job applicants in coding tests

Rachele Mello
Department of Software Engineering and Management
University of Gothenburg
Gothenburg, Sweden
rachelemello@gmail.com

***Abstract*—Several software companies use some sort of competitive programming to screen job applicants. In this study, the factors present in job applications are analyzed to find possible predictors of candidates' scores in competitive programming tests. Non parametric statistical tests are used and a logistic model is built and evaluated.**

I. Introduction

Competitive programming is a "mind sport" in which participants solve well-defined algorithmic problems by writing computer programs under specified limits (Halim and Halim 2013). This discipline is rather popular among programmers, both as a leisure activity and as a way to develop stronger programming skills. Additionally, some forms of competitive programming are also used by software companies in talent recruitment (McDowell 2011, Jokela 2017).
In the case of Google, for example, the first stage of the hiring process consists of a phone interview where candidates are asked to code solutions to defined algorithmic problems, while thinking aloud[1]. Other companies use services that automate this assessment by providing a platform where candidates need to code the solution to problems similar to the ones in competitive programming within a given time limit. These services then automatically evaluate solutions based on a set of parameters (e.g. correctness, performance, complexity) and report the candidates' scores to the company. The numerous companies providing this sort of service (e.g. Codility[2], InterviewZen[3], Tests4Geeks[4], HackerRank[5]) suggests that this method for screening applicants is widely used in software companies today.

Whether this initial screening, aided by competitive programming, is done manually or automatically, it is a cost for the recruiting company. First of all, in both cases a pre-screening needs to be done to select the candidates for the coding test/interview. In the case of Google, it is clear how having an interviewer prepare, conduct, and evaluate such screening costs significant amount of resources to the company. Even if it is a cheaper option, companies still need to pay a fee to send programming tests to candidates through the services mentioned before, and the output of these tests still needs to be evaluated manually.

Apart from the costs, another issue with this kind of talent recruitment process is that, especially when there is an overwhelming amount of applications for a certain position, the initial pre-screening cannot be carried out very meticulously. This might result in the exclusion of valuable candidates from the hiring process.

Thus, a more cost-effective way to select interviewees is desirable. This study evaluates whether it is possible to effectively predict the results of coding tests, given only a candidate's job application documents (i.e. resume and cover letter). A positive result would allow organizations to decrease the costs of recruitment by providing a much quicker and effective way to pre-screen candidates, or even skip entirely the recruitment step involving competitive programming, directly selecting the interviewees.

This study brings a technical contribution with methodological guidelines that companies could employ when screening applications for software developer positions, as well as a scientific contribution with the extension of the current research on methods and tools to evaluate candidates and validating the existing studies on the characteristics of highly performant software developers.

II. Research Question

RQ: Are there any factors present in job application documents that can predict the candidate's performance in programming tests?

III. Background and related work

A. Improving recruitment

In literature we can find several solution approaches to the problem of making recruitment of developers easier and more

[1] https://careers.google.com/how-we-hire/interview/
[2] https://codility.com/
[3] https://www.interviewzen.com/
[4] https://tests4geeks.com/
[5] https://www.hackerrank.com/

effective.

Sarma et al. (2016) provide a tool, named Visual Resume, that aggregates activity traces of developers across different types of contributions and repositories into a single developer profile, making them easier to be used in the hiring process.
How GitHub traces are used in the hiring process has also been studied by Marlow and Dabbish (2013). Specific cues on contributors' profiles are seen by employers as indicators of technical skills, motivation and values. Some of the identified cues (e.g. side projects) can be present in resumes as well, making the study relevant for my research.
In both of these studies, the researchers' starting point is that developers' online contributions are used more and more by managers in their hiring decisions, but this choice is not challenged. A correlation between certain characteristics of developers' online contributions and their skills is inferred but not proven.

McCuller (2012) analyses the whole recruitment and hiring process of software engineers, giving guidelines to organizations. Particularly relevant for this study are his definitions of "good" and "bad" resumes and what to specifically look for in them.
In this case as well, the cues that the writer suggests to pay attention to come from experience rather than empirical evidence.

Another tool to facilitate the hiring process is designed by Menon and Rahulnath (2016). Their tool automates the eligibility check and aptitude evaluation of job applicants by analyzing their resumes and social media profiles.
In this study a system is built, using machine learning and regression techniques, which ranks the candidates in the order of their compatibility score to the job position. The system accuracy is tested against the ranking given by an experienced recruiter.
Among the indicators considered in the study there are some linguistic ones that can be applied to curricula as well.

B. Characteristics of good software developers

Different studies have tried to identify the characteristics of good software developers and what employers seek in them. These studies have been useful in determining the factors to analyze in the job application documents for this study.

Wynekoop and Waltz (2000) propose a methodology for building a model of the personality traits of top performing developers. They then conduct a pilot study on students, where the ones identified as top performing developers are subjected to a personality test to evaluate if they possess the identified traits.

The purpose of the study conducted by Ahmed et al. (2012) is to find whether the soft skills that employers look for in software developers vary from culture to culture. The result is that culture does not generally have an impact, but the paper presents a collection of desired soft skills in programmers.

These two studies, among others, suggest that exceptional developers possess resembling personality traits and soft skills. These traits and skills might appear in the style of writing or the choice of information to include in the job application documents.

Clark et al. (2003) investigate the differences in experienced and novice IT professionals, aiming at providing guidelines for selecting those novice job applicants with the potential of becoming expert developers. Their results show that positive extraversion is found in top performing experienced IT professionals, while negative extraversion is found in IT students with high GPA. Therefore, their study suggests that extraversion is more important than academic performance in the long run in the IT field.

There also exist several posts on technical blogs and online magazines in gray literature that try to identify the characteristics of good programmers, such as James (2008).

C. Related works

A few studies have approached the problem in a similar way as this paper does.

Evans and Simkin (1989) studied predictors of academic performance in programming courses. Cegielski and Hall (2006) conducted a study on whether theoretical believes, cognitive abilities and personality of software developers related to their performance in object-oriented programming tests. Bachrach (2015) studied how social media profiles relate to perceived job-suitability, finding profile components as well as education, skills and demographic traits to be predicting factors. Douglas et al. (2013) analyze the situation in a company, developing an algorithm that can predict employees' performance given their biographical information and entry test scores.

However, the difference between this study and the ones mentioned above is that they either try to predict something different, such as job-suitability, academic or job performance, or/and they use different basis than what is available in job application documents to study the presence of predictors.
No previous work could be found that studies the possibility of predicting performance at programming tests given a candidate's submitted curriculum.

D. Applied statistics in Software Engineering

Applied statistics is useful in Software Engineering experimentation.
Both Juristo and Moreno (2013), and Emam and Carleton

(2004) agree on both the importance and lack of experimentation and proper use of statistical methods in this field.
Juristo and Moreno (2013) observe that nowadays the Software Engineering research is not much based on rigorous and objective data but more on opinions and anecdotal experience.

This study aims at contributing to the Software Engineering research through an empirical analysis on the contents of job applications, as opposed to the current practice in recruitment which is based on subjective criteria.

IV. Research Methodology

In order to answer the research question, I conducted a study on the hiring process of software developer interns in Opera Software.

Opera Software is a Chinese-owned mid-sized software company founded in Oslo, Norway, and with worldwide offices. The company develops and markets web browsers for both desktop and mobile platforms, reaching more than 350 million users.
Opera's Gothenburg and Linköping offices focus mainly on the development of "Opera for Android" and "Opera Mini" browsers. Every summer, a total of approximately 8 engineering interns are hired to join the teams of software developers in the two offices. Hundreds of students apply to these positions every year, and after an initial screening done by the human resources department and the team leaders, a limited number of applicants is assessed through some coding challenges on the Codility platform. The candidates who best perform in these challenges are finally invited to an on-site interview, which constitutes the final step of the hiring process.

The study conducted on the hiring process was split into four phases: data collection, data preparation, analysis and identification of predictors, and evaluation.
In the first phase, the job application documents and results from the programming tests are collected.
In the second phase, the job application documents are described by a series of attributes. Attributes are identified from previous studies and articles, input from the employees involved in the selection of candidates. Some attributes are included due to their easy availability, to maximize the chances of finding suitable predictors.
In the third phase, dependency of the test from the identified parameters is tested employing different statistical tools, and part of the data is used to build a binomial logistic regression model.
Finally, the predictive ability of the model is evaluated through the calculation of its accuracy, and visualized through a ROC plot.

A. Data collection

For this study, both qualitative and quantitative data has been collected from Opera's internships recruitment process of spring 2017.
Qualitative data is constituted by the documents submitted by each candidate in their job application. Opera requires candidates to submit a curriculum vitae and (optionally, but encouraged) a cover letter. No particular format is imposed for these documents. The results of the programming assessment tests constitute quantitative data: each test receives an overall score, and every one of the three tasks in the test is scored on both correctness and performance, each represented by a percentage.

The total number of applicants for the developer summer jobs was 492 (283 applicants for the positions in Gothenburg and 209 applicants for the positions in Linkoping). Of these, 93 candidates (58 in Gothenburg, 35 in Linkoping) were selected by the human resources department and the team leaders to be taken to the next step of the hiring process and receive a programming test. An additional 25 candidates were selected randomly among the remaining, to provide a more generalized sample. In total, 118 candidates received an invitation from Opera to complete a programming test on the platform Codility.

Of these 118 candidates who received the test from Opera, 33 did not take the test. Of the 85 tested candidates, 13 could not be considered for the study for different reasons:

- in 4 cases substantial similarities with previous submissions or with solutions found online were detected by the platform;
- 3 candidates started the test but did not attempt it (exclusion criteria: less than 15 minutes of effective time spent on the test and a score of 0);
- 6 candidates submitted their application documents in Swedish.

Therefore, the sample available for this study consists in 72 job applications and corresponding test results.

B. Data Preparation

1) Input Attributes: The input attributes were chosen based on previous studies, technical blog posts and books on the desirable qualities of software developers, and input from employees involved in the hiring process in the company. Some miscellaneous and descriptive input parameters were included as they are easily attainable through text processing and analysis software or manual screening.

The input attributes and their variable type are shown in Table I, together with notes on the used scale, reference to previous studies that take these attributes into account, and whether or not they have been reported by Opera's hiring managers and HR to be part of their selection criteria.

Attribute	Variable type	Scale / Notes	Derived from previous work	Used by company
Cover letter	Binary	0 = not submitted, 1 = submitted		X
Gender	Binary	0 = male, 1 = female	[12]	
Level of studies	Binary	0 = bachelor, 1 = master	[12]	X
Photo	Binary	0 = not present, 1 = present		
Pages CV	Numerical		[3]	X
Vocabuary density	Numerical		[4]	
Words/Page	Numerical		[3]	X
GitHub/BitBucket	Binary	0 = not present, 1 = present	[1], [2]	X
Programming languages	Numerical			
Personal projects	Binary	0 = not present, 1 = present	[1], [2]	X
Current field studies	Categorical	CS = Comp. science, SE = Software eng., O = other	[3], [12]	X
Education outside Sweden	Binary	0 = no, 1 = yes		
Languages	Numerical		[1]	
Experience as developer	Binary	0 = no, 1 = yes	[3]	X
Student associations (years)	Numerical		[7]	X
Scholarship	Binary	0 = no, 1 = yes	[7]	
Own company	Binary	0 = no, 1 = yes	[3]	
Teaching/lab assistant	Binary	0 = no, 1 = yes		
References	Binary	0 = no, 1 = yes		
Android	Binary	0 = no, 1 = yes		X
Algorithm	Binary	0 = no, 1 = yes		X
Selected	Binary	0 = no, 1 = yes		

TABLE I
INPUT ATTRIBUTES

The next paragraphs provide the definition of the criteria and tools used for those attributes that require further explanation.

The attribute ”Level of studies” describes whether an applicant is currently enrolled in a bachelor’s or master’s program. For applicants pursuing a comprehensive 5-years university program (e.g. the Swedish ”civilingenjör”), the enrollment year is considered to assign this variable: 0-3 years from enrollment is considered as bachelor student, 4-5 years as master student. In case the applicant reports delay in their studies, or taking a break, these are considered as well.

The text processing and analysis software Voyant tools[6] is used to extract the information regarding the vocabulary density and the total words in the curricula. The attribute ”Vocabulary density” is defined as the ratio between the number of unique words and the total words in a text. The attribute ”Words/Page” is the ration between the total amount of words and the number of pages of the candidate’s curriculum.

To count the ”Programming languages” present in each curriculum, an online list of all notable existing programming languages is used as a reference[7]. Languages listed under ”Skills”, ”Programming languages” or other similar sections of the curricula are considered. As some candidates report a level of proficiency for each of the programming languages while others do not, it was decided to count all mentioned languages.

”Languages” refer to natural languages a candidate claims to know, at least at an elementary level. For candidates that do not mention any language in their curricula, the value of ”1” was recorded. In one instance, the candidate reported knowing ”sign language”, which was considered towards this count.

As the vast majority of candidates are either students of computer science or software engineering, the ones who are not are grouped in the general category ”Other” for ”Current field of studies”.

The variable ”Projects” indicates whether or not a candidate describes at least one programming project (academic or personal) in their curriculum.

Some of the candidates include a link to their GitHub or BitBucket profiles in their resume. This is indicated by the variable “GitHub/BitBucket”.

The variable ”References” indicates whether or not a candidate includes references in their curriculum. The

[6]https://voyant-tools.org/
[7]https://en.wikipedia.org/wiki/List_of_programming_languages

classical line ”References will be provided upon request.” is not considered for this variable.

The variables ”Android” and ”Algorithms” refers to whether or not the applicant mentions these in their curriculum, either as an interest, or something they have studied or worked with (both academically or not).

Finally, the variable ”Selected” is true for candidates that were selected by the HR or team leaders in the company, and false for those who were randomly chosen among the discarded ones to be tested anyways for this study.

2) Output Attribute: The output attribute consists in the overall result of the test. The test used by Opera in the recruitment is made of 3 tasks. Each task is evaluated on correctness and performance, both represented as a percentage. A score on a scale 0-100 is given to each task and an overall score for the test is given on a scale 0-300 by combining the scores of the 3 tasks.
The output attribute and its variable type are shown in Table II.

Attribute	**Variable type**
Test score	Numerical

TABLE II
OUTPUT ATTRIBUTE

3) Analysis and validation sets: The data set of 72 job applications and test results described by the input and output attributes has been divided into two:

- an analysis set, containing 57 instances (80% of the total);
- a validation set, containing 15 instances (20% of the total).

The independent variable “Selected” has been excluded from these data sets.
The division of the instances between the two sets has been performed randomly, by using a random number generator to pick candidates out of the complete data set.

C. Data Analysis

The correlation of each single independent variable to the dependent variable “Test score” is studied by means of an appropriate statistical test depending on the nature of the variables and their distribution. These tests are performed on the complete data set of 72 observations.

The analysis data set is used to construct a binomial logistic regression model in which the output variable is whether a candidate scores over 200 points in the test. Binomial logistic regression models are used to predict a binary dependent variable given a set of predictors.
The threshold of 200 points over 300 possible is chosen to match the threshold the hiring managers of Opera Software adopt when selecting candidates to bring to the next recruitment stage.

D. Evaluation

The data from the validation set is used to calculate the reliability of the constructed model.
The regression model is applied to the data and the predicted booleans of whether a candidate will score higher than 200 points in the test are confronted against the actual results of the tests. The accuracy of the model, as well as the ROC curve and AUC value are used to assess its predictive ability.

V. RESULTS

A. Collected data

The tables containing the collected data can be found in the appendix.

The Shapiro-Wilk normality test on the dependent variable “Test score” yields a p-value of 0.2739. Given that the p-value is larger than 0.05, the Shapiro-Wilk test’s hypothesis that the data follows the normal distribution can not be rejected. However, the histogram and normal Q-Q plot of “Test score” shown in Figure 1 suggest that the underlying distribution is light tailed.

Therefore, it is concluded that the data does not follow the normal distribution.

B. Correlation

As it can not be assumed that the data follows the normal distribution, non parametric tests are chosen to evaluate correlations between each independent variable and the test score.

1) Correlation between numerical variables and test score: The non parametric test Spearman’s Rank-Order correlation has been used to determine the presence, strength and direction of monotonic correlations between the dependent variable ”Test score” and each of the numerical independent variable. The general null hypothesis being tested is that true rho = 0, i.e. no monotonic correlation exists.
Table III shows the results of the Spearman’s Rank-Order correlation test on the 6 numerical attributes.

2) Correlation between binary variables and test score: The Mann-Whitney-Wilcoxon test does not assume normality and can be used to determine whether the population distributions are identical. The general null hypothesis being tested is that presenting one or the other characteristic of each binary attribute does not influence the job applicant’s score at the programming test.
The results of the Mann-Whitney-Wilcoxon tests on the 15

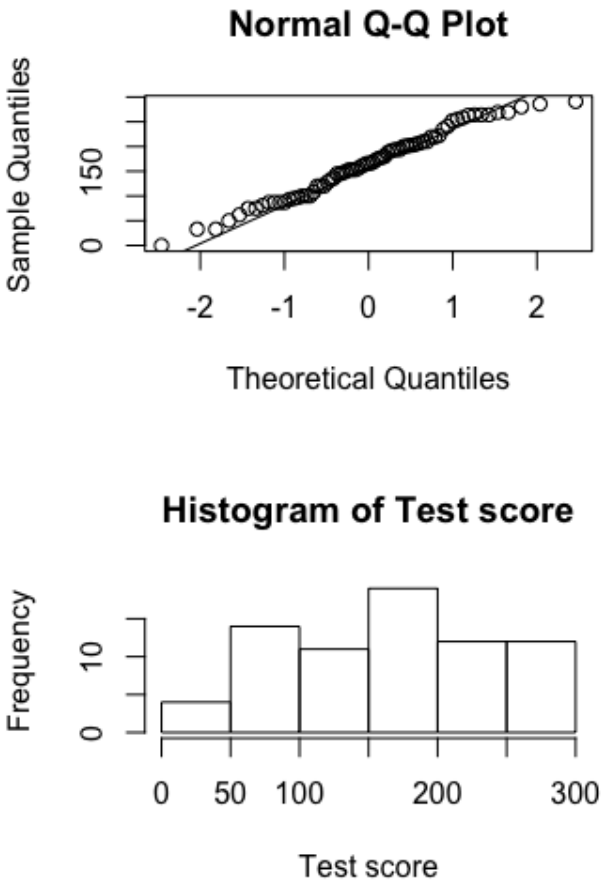

Fig. 1. Normal Q-Q plot and histogram of dependent variable "Test score"

Input attribute	S	p-value	rho
Pages CV	73571.0	0.1241	-0.1828972
Vocabulary density	62174.0	0.9976	3.538229E-4
Words/Page	55506.0	0.3685	0.1075656
Programming languages	63163.0	0.8968	-0.01555365
Languages	73746.0	0.1183	-0.1857089
Student associations	51366.0	0.1435	0.1741214

TABLE III
SPEARMAN'S RANK-ORDER CORRELATION TEST RESULTS

binary attributes are reported in Table IV.

3) Correlation between categorical variable and test score: To investigate the influence of the field of studies on the test score, the Chi-squared test of independence is used. The null hypothesis being tested is that the test scores are independent of the candidate's field of studies.
Table V shows the result of the test.

Input attribute	W	p-value
Cover letter	512.5	0.2716
Photo	668.0	0.6133
References	264.5	0.7527
Level of studies	649.5	0.8281
Education outside Sweden	497.0	0.7772
Scholarship	316.0	0.9286
Teaching/lab assistant	411.0	0.8244
GitHub/BitBucket	413.0	0.3456
Projects	534.5	0.5324
Experience as developer	590.5	0.6127
Own company	131.5	0.9216
Android	658.0	0.8024
Algorithms	638.0	0.6432
Gender	336.5	0.672
Selected	638.0	0.6432

TABLE IV
MANN-WHITNEY-WILCOXON TEST RESULTS

Input attribute	X-squared	df	p-value
Field of studies	92.387	104	0.7854

TABLE V
CHI-SQUARED TEST OF INDEPENDENCE RESULT

C. Binomial logistic regression model

1) Model fitting: The binomial logistic regression model fitted with the data from the analysis data set is shown in Figure 2.

```
Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -1.75068   12.40891  -0.141   0.8878
`Cover letter`                 -1.06911    1.45066  -0.737   0.4611
`Pages CV`                     -0.87461    1.51587  -0.577   0.5640
Photo                          -3.27037    2.13251  -1.534   0.1251
References                      2.46708    2.66877   0.924   0.3553
`Vocabuary density`            -0.49139   12.57278  -0.039   0.9688
`Words/Page`                    0.02816    0.01700   1.656   0.0977 .
`Level of studies`             -5.43295    2.75632  -1.971   0.0487 *
`Current field of studies`0     7.69664    2.96762   2.594   0.0095 **
`Current field of studies`SE    1.45223    1.62563   0.893   0.3717
`Education outside Sweden`      8.51798    3.95688   2.153   0.0313 *
Scholarship                    -1.36637    1.79305  -0.762   0.4460
`Student associations (years)`  1.37313    0.93486   1.469   0.1419
`Teaching/lab assistant`       -5.18182    2.94393  -1.760   0.0784 .
`GitHub/BitBucket`             -1.82293    1.71071  -1.066   0.2866
`Programming languages`         0.44771    0.33069   1.354   0.1758
Projects                        0.92249    1.58780   0.581   0.5613
`Experience as developer`      -1.13769    1.58879  -0.716   0.4739
`Own company`                   2.63190    2.41705   1.089   0.2762
Android                        -4.88874    2.32469  -2.103   0.0355 *
Algorithms                      1.64386    1.76062   0.934   0.3505
Gender                        -21.06004 2725.50991  -0.008   0.9938
Languages                      -1.38314    1.07854  -1.282   0.1997
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig. 2. Summary of the coefficients of the binomial logistic model

2) Assessing the model: The model is applied to the observations from the validation data set to predict whether

a candidate will score over 200 points in the programming test. The decision boundary of 0.5 is used to transform the outputted probabilities of the model into a binary response.
Table VI shows the outputted probability, the predicted and actual outcome of a candidate performing over 200 points at the programming test, for each observation in the validation data set.

Probability	Predicted outcome	Actual outcome
5.401709E-12	0	1
8.537966E-4	0	0
0.8386135	1	1
0.8123795	1	0
0.9881359	1	0
0.006419624	0	0
0.6162502	1	1
2.923078E-8	0	0
0.01987276	0	0
0.9797599	1	0
0.8390977	1	0
1.987217E-4	0	0
0.2058407	0	1
2.183347E-10	0	0
0.6479676	1	0

TABLE VI
PREDICTED VS ACTUAL HIGH PERFORMANCE AT TEST

The accuracy of the model on the validation set is 0.53, calculated as the ratio between the correct predictions (sum of true positives and true negatives) and the total population.

Figure 3 shows the ROC (Receiver Operating Characteristic) curve generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The AUC (i.e. Area Under the Curve) value is 0.45.

VI. Analysis and Discussion of Results

A. Analysis

1) Correlations: As shown in Table IV, none of the results of the Mann-Whitney-Wilcoxon tests reports a p-value lower than 0.05. Therefore, in none of these tests can the null hypothesis be rejected, meaning that the test scores of the two groups for each binary variable are identical populations. In practice, this result indicates that none of the studied binary variables can be used as a statistically significant predictor of the test score. Master students do not score higher than bachelor students, men do not score higher than women, candidates including a cover letter do not score higher than those who do not, etc.

Similarly, all the results of the Spearman's Rank-Order correlation tests report a p-value higher than 0.05, as reported in Table III, and none of the null hypotheses can be rejected.

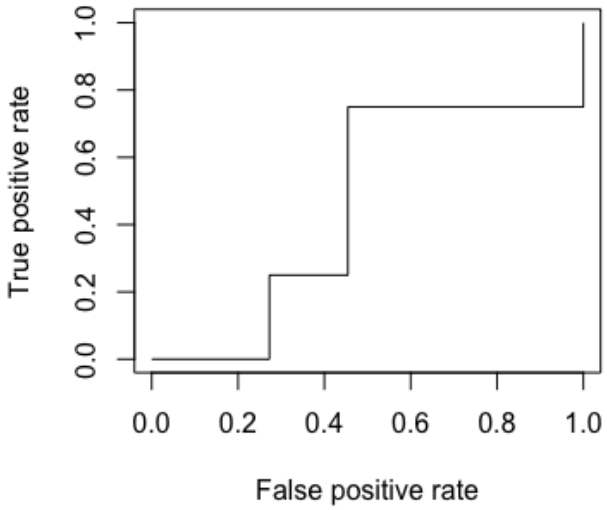

Fig. 3. ROC plot of the binomial logistic model

Therefore, there is no statistically significant correlation between any of the numerical independent variables and the dependent variable "Test score".

The field of studies of a candidate is also not useful in predicting the score of the test, as the p-value of the Chi-quared test is higher than 0.05 as reported in Table V.

Overall, none of the attributes taken into consideration in this study can alone be considered a predictor of the score at the programming test.

2) Logistic model: Most of the independent variables of the logistic model are not statistically significant ($p > 0.05$ as reported in Figure 2), as it was foreseeable from the results of the correlation tests on the single variables.
The "other" field of studies has the lowest p-value, suggesting a strong relationship between candidates who do not study Software Engineering or Computer Science and their high performance at the programming test. The positive coefficient of the predictor indicates that, all other variables being equal, the candidate who does not study CS or SE has a higher probability of scoring over 200 points in the test.
Similarly, having had University education outside Sweden improves the probability of scoring high at the test. On the other hand, being a master student and reporting experience or interest in Android lowers the chance of scoring over 200 points.

However, the accuracy of the model is very low. 53% accuracy means that flipping a coin to predict whether a

candidate will score high at the test would have basically the same accuracy as applying this model to achieve the prediction.
This can also be seen from the ROC plot reported in Figure 3: the line plotting the TPR against the FPR runs alongside the axes diagonal, meaning that trying to increase the TPR increases the FPR by a similar value, as they go hand-in-hand. In fact, the AUC value is 0.45, while for a good prediction model we seek values closer to 1.

The accuracy of the logistic model could be dependent on the way the data was split into the analysis and validation sets and a more precise value could be achieved using cross validation. However, given the low accuracy, seeking a more precise value is not relevant.

B. Discussion

The results and their analysis give the answer to the research question: this study could not find any factors present in the job application documents that can predict the candidate's performance in programming tests.
This does not imply that no predictors of coding tests performance is present in the resume and cover letter of an applicant. A different approach could identify them, as will be discussed later in Section VI-D, "Future work".

It is interesting to notice that when comparing candidates that have been pre-screened and selected by the HR and team leaders at Opera Software to the ones that were randomly selected among the discarded ones, we can not reject the hypothesis that they come from identical populations in relation to their scores at the test.
This result indicates that there is no need for companies that use competitive programming as a mean to select candidates in the early stages of recruitment to spend man hours on the screening of application documents: the same results can be achieved by selecting a random sample of applicants to test, and time can be more efficiently spend on reviewing the applications and quality of solutions of the smaller group of high scorers.

From an ethical point of view, it is arguable whether we should at all seek an algorithm able to filter job applicants. While on one side some might argue that being evaluated by a machine feels impersonal and unappreciative of the efforts one puts into creating a good resume and cover letter, others might claim that a busy manager would also just skim through a job application, and that automating the process might increase fairness as human biases are removed.

C. Threats to validity

A threat to construct validity is represented by the particular choice of input attributes. Different attributes might have shown correlation with the result of the test or improved the accuracy of the logistic model, potentially changing the answer to the research question. To reduce this risk, as many attributes a possible, given the limitations of this study, were included, with consideration of what could be extracted from previous literature and the company help (as reported in Table I).

The chosen statistical tests can also represent a threat to construct validity. Tests for normality were employed to choose appropriate methods. Additional and alternative possibilities for the analysis of the data are discussed in the next section.

Even though the process of compiling the data set by reading the job applications and annotating the different attributes was done carefully and meticulously, it is possible that some mistakes were made, compromising the reliability of the findings. It would be advisable to have a partner researcher validate the data.

Finally, in regards to external validity, it can be argued whether the results (or lack of thereof) of this study can be generalized to other processes of recruitment of software developers that use competitive programming. The candidates in this study were all students, while companies would more generally deal with the recruitment of professionals, who report different information in their resume (e.g. previous roles and years of experience).

D. Future work

For future work, my suggestion is to try and find more input attributes that can describe the application documents. The better the candidate's job application is described, the higher the chance of finding predictors among the attributes.
Moreover, it is valuable to look at the interaction among the variables, as the unique characteristics of top developers are probably more complex than simple factors alone.
Finally, the use of Bayesian statistics could be investigated, as it is a valid alternative to logistic regression.

VII. Conclusion

The study on the recruitment process of developer interns in Opera Software could not highlight any factors present in candidates' job application documents that serve as suitable predictors to the candidates' performance at the programming tests used as screening tool. Neither single attributes showed any correlation to the test results, nor a high score could be effectively predicted through the construction of a logistic regression model. Notably, no difference in test scores was found between candidates that had been selected by Opera Software during the initial screening of the applications, and candidates that were discarded by the company but tested for the sake of this study.

Given this last result, software companies which use competitive programming as a mean to screen the job applicants are advised to use the test as a first filter and only then perform a manual selection of the top performing candidates, based on the application documents and quality of solutions.
Ideally, the programming test would be sent to all applicants, but, if this proves to be expensive, a random subset of candidates can be chosen to be assessed through competitive programming. An accurate evaluation of the top scorers can then be performed manually. As this will be a much smaller number than the total applicants (in the case here presented only 30% of the tested candidates scored above 66%), selecting interviewees will be considerably cheaper at this point.

Finally, some of the examined factors are considered by the Software Engineering research to be characteristics or cues for top developers. As such factors did not correlate to high scores in the administered programming test, more research should be done on whether or not top developers actually hold such characteristics, as well as studying whether skills in competitive programming are actually correlated to high job performance and should be used as a tool to select candidates.

ACKNOWLEDGMENT

The author would like to express her gratitude to Jan-Philipp Steghöfer for his guidance throughout this project, and to Hanna Björk for her help in navigating through the recruitment process and tools at Opera Software.

REFERENCES

[1] Sarma, Anita, et al. ”Hiring in the global stage: Profiles of online contributions.” Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on. IEEE, 2016.

[2] Marlow, Jennifer, and Laura Dabbish. ”Activity traces and signals in software developer recruitment and hiring.” Proceedings of the 2013 conference on Computer supported cooperative work. ACM, 2013.

[3] McCuller, Patrick. ”How to Recruit and Hire Great Software Engineers: Building a Crack Development Team.” Apress, 2012.

[4] Menon, Vishnu M., and H. A. Rahulnath. ”A novel approach to evaluate and rank candidates in a recruitment process by estimating emotional intelligence through social media data.” Next Generation Intelligent Systems (ICNGIS), International Conference on. IEEE, 2016.

[5] Wynekoop, Judy L., and Diane B. Walz. ”Investigating traits of top performing software developers.” Information Technology People 13.3 (2000): 186-195.

[6] Ahmed, Faheem, et al. ”Soft skills requirements in software development jobs: a cross-cultural empirical study.” Journal of systems and information technology 14.1 (2012): 58-81.

[7] Clark, Jan Guynes, Diane B. Walz, and Judy L. Wynekoop. ”Identifying exceptional application software developers: A comparison of students and professionals.” (2003).

[8] James, Justin. ”10 traits to look for when you’re hiring a programmer.” (2008).

[9] Evans, Gerald E., and Mark G. Simkin. ”What best predicts computer proficiency?.” Communications of the ACM 32.11 (1989): 1322-1327.

[10] Cegielski, Casey G., and Dianne J. Hall. ”What makes a good programmer?.” Communications of the ACM 49.10 (2006): 73-75.

[11] Bachrach, Yoram. ”Human judgments in hiring decisions based on online social network profiles.” Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on. IEEE, 2015.

[12] Augusto, Douglas A., Heder S. Bernardino, and Helio JC Barbosa. ”Predicting the performance of job applicants by means of genetic programming.” Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on. IEEE, 2013.

[13] Khaled El Emam, Anita D. Carleton, ”Applications of statistics in software engineering”, Journal of Systems and Software, Volume 73, Issue 2, 2004, Pages 181-182, ISSN 0164-1212.

[14] Juristo, Natalia, and Ana M. Moreno. ”Basics of software engineering experimentation.” Springer Science Business Media, 2013.

[15] Halim, Steven, and Felix Halim. ”Competitive Programming 3.” Lulu Independent Publish, 2013.

[16] McDowell, Gayle Laakmann. ”Cracking the coding interview.” CarrerCup, 2011.

[17] Jokela, Juho. ”Evaluating and measuring the adequacy of a programming job applicant: Using code tests as a method of evaluation.” 2017.

APPENDIX A - Complete data set

ID	Cover letter	Pages CV	Photo	References	Vocabuary density	Words/ Page	Level of studies	Current field of studies	Education outside Sweden	Scholarship	Student associations (years)	Teaching/ lab assistant
01	0	4	0	1	0.49	261.50	1	SE	1	0	0	1
02	1	2	0	0	0.63	259.50	1	O	1	0	3	1
03	1	1	1	0	0.73	154.00	0	SE	0	0	1	0
04	0	2	0	0	0.54	201.00	0	SE	0	0	0	0
05	1	1	0	0	0.72	244.00	1	CS	1	1	0	0
06	0	3	0	0	0.57	132.00	0	CS	0	0	0	0
07	0	2	1	0	0.65	151.50	1	CS	0	0	0	0
08	1	2	1	0	0.61	215.50	1	CS	1	0	2	0
09	0	1	1	0	0.58	454.00	1	CS	1	0	0	0
10	1	1	0	0	0.72	224.00	0	CS	0	0	0	0
11	0	1	1	1	0.68	340.00	0	O	0	0	0	0
12	1	2	1	0	0.62	271.00	1	O	0	0	0	0
13	0	2	0	0	0.59	182.00	1	CS	1	1	0	0
14	0	2	0	0	0.57	388.50	1	CS	0	0	2	1
15	0	1	0	0	0.69	228.00	1	SE	1	0	0	0
16	0	1	1	0	0.68	269.00	1	O	1	0	0	0
17	0	3	0	0	0.55	157.67	0	O	0	0	0	0
18	0	2	1	0	0.63	180.50	0	CS	0	0	2	0
19	0	2	0	0	0.67	238.00	1	O	0	0	0	0
20	0	2	0	0	0.61	169.50	0	SE	0	0	2	0
21	1	2	1	0	0.58	234.00	1	CS	0	0	2	0
22	0	2	0	1	0.63	199.50	1	CS	0	0	0	0
23	0	2	1	0	0.63	236.00	1	CS	0	0	1	0
24	1	2	0	0	0.84	62.50	0	CS	0	0	0	0
25	0	2	1	0	0.63	201.50	0	CS	0	0	0	0
26	0	2	0	0	0.60	186.00	0	CS	0	0	0	0
27	1	2	1	0	0.61	187.00	0	CS	0	1	1	0
28	1	1	0	0	0.68	106.00	1	CS	0	0	0	1
29	1	1	0	0	0.68	106.00	1	CS	0	0	0	1

APPENDIX A - Complete data set

ID	GitHub/ BitBucket	Programming languages	Projects	Experience as developer	Own company	Android	Algorithms	Gender	Languages	Selected	Test score	Test > 200
01	0	10	0	1	0	1	0	0	3	0	100	0
02	0	7	0	1	0	1	0	0	3	0	258	1
03	0	3	0	0	0	0	0	0	3	0	176	0
04	1	4	0	0	0	0	0	0	2	0	193	0
05	0	5	0	0	0	1	1	1	4	1	120	0
06	0	10	0	1	0	0	1	0	3	1	120	0
07	0	4	0	0	0	0	0	0	3	0	81	0
08	1	10	0	1	0	1	1	1	5	1	264	1
09	1	4	0	1	0	1	0	0	1	0	222	1
10	0	0	1	0	0	0	0	0	2	0	156	0
11	0	3	0	0	0	1	0	0	4	0	264	1
12	0	3	0	0	0	1	1	0	5	1	100	0
13	0	5	0	1	0	0	1	0	3	1	235	1
14	1	9	0	1	0	0	1	0	2	1	87	0
15	0	0	0	1	0	0	0	0	3	0	264	1
16	0	2	0	1	0	1	0	0	4	0	87	0
17	1	5	1	1	1	1	0	0	3	0	255	1
18	0	6	1	0	1	1	0	0	3	0	33	0
19	0	4	0	0	0	0	1	0	3	1	147	0
20	0	6	0	1	1	1	0	0	3	0	286	1
21	0	7	0	0	0	0	1	0	1	1	291	1
22	1	5	0	0	0	0	0	0	2	0	163	0
23	0	7	0	0	0	0	1	0	3	1	128	0
24	1	7	0	0	0	0	0	0	1	0	108	0
25	0	6	0	0	0	0	1	1	3	1	173	0
26	0	4	0	1	0	1	0	0	2	0	178	0
27	0	6	0	0	0	0	1	0	2	1	269	1
28	0	0	0	0	0	1	0	0	2	0	194	0
29	0	0	0	0	0	1	0	0	2	0	194	0

APPENDIX A - Complete data set

ID	Cover letter	Pages CV	Photo	References	Vocabuary density	Words/ Page	Level of studies	Current field of studies	Education outside Sweden	Scholarship	Student associations (years)	Teaching/ lab assistant
30	1	3	0	0	0.64	127.00	1	SE	1	0	1	0
31	1	2	0	0	0.51	296.00	1	CS	0	0	3	1
32	1	1	0	0	0.61	293.00	0	CS	0	0	0	1
33	1	2	1	1	0.67	180.50	0	CS	0	0	1	0
34	1	1	1	0	0.70	207.00	1	O	0	0	0	0
35	0	1	1	1	0.60	379.00	1	CS	0	1	3	0
36	1	1	1	1	0.87	121.00	0	CS	0	0	0	0
37	1	3	0	0	0.60	170.00	0	SE	0	0	2	0
38	1	2	0	0	0.61	115.50	0	SE	0	0	0	0
39	0	2	0	0	0.54	227.00	1	CS	1	0	3	1
40	1	2	1	0	0.56	277.50	1	CS	0	0	2	0
41	0	4	0	0	0.49	230.00	1	SE	1	1	0	0
42	0	2	0	0	0.66	186.50	1	CS	0	0	4	0
43	0	2	0	0	0.71	130.50	0	SE	0	0	0	0
44	0	2	1	0	0.56	225.00	1	CS	0	0	0	0
45	0	1	1	0	0.68	211.00	1	CS	0	0	0	0
46	1	2	0	0	0.60	175.50	0	SE	1	0	1	0
47	0	2	0	0	0.66	115.00	1	CS	1	1	0	0
48	0	4	1	0	0.55	171.75	1	O	0	0	4	0
49	0	3	0	1	0.54	221.33	1	CS	1	0	0	0
50	1	2	0	0	0.55	231.50	1	SE	0	0	1	0
51	0	1	1	1	0.60	508.00	1	CS	0	0	0	0
52	1	2	1	0	0.58	233.00	0	CS	0	0	0	0
53	0	1	0	0	0.73	248.00	1	CS	0	0	0	1
54	0	2	0	0	0.49	184.50	0	O	0	0	0	1
55	0	2	1	0	0.61	201.50	0	CS	0	0	0	0
56	1	3	1	0	0.58	157.00	1	SE	0	0	0	0
57	0	2	0	0	0.62	191.00	0	O	0	0	2	1
58	0	1	0	0	0.77	81.00	0	CS	0	0	0	0

APPENDIX A - Complete data set

ID	GitHub/ BitBucket	Programming languages	Projects	Experience as developer	Own company	Android	Algorithms	Gender	Languages	Selected	Test score	Test > 200
30	0	7	1	0	0	1	0	0	4	0	75	0
31	1	8	1	0	0	1	1	0	2	1	200	0
32	1	8	0	0	0	0	1	0	2	1	219	1
33	0	6	1	1	0	0	0	0	1	0	269	1
34	0	7	0	0	0	0	0	0	1	0	203	1
35	1	6	0	1	0	1	0	0	3	0	153	0
36	0	8	0	1	0	0	0	1	1	0	188	0
37	0	6	1	0	0	1	0	0	3	0	153	0
38	0	8	0	1	0	0	0	0	3	0	33	0
39	0	7	0	1	0	0	0	0	3	0	202	1
40	0	10	0	0	0	0	1	0	2	1	145	0
41	0	7	1	1	0	1	0	0	2	0	210	1
42	0	7	0	0	0	0	0	0	3	0	263	1
43	0	5	0	0	0	0	0	0	3	0	219	1
44	0	7	0	1	0	1	1	1	2	1	95	0
45	0	4	0	1	0	0	1	1	2	1	150	0
46	0	4	1	0	0	1	0	1	4	0	166	0
47	0	4	0	1	0	0	0	0	1	0	75	0
48	1	5	1	0	0	1	0	0	3	0	133	0
49	0	3	1	0	0	0	0	0	4	0	0	0
50	1	5	0	0	0	0	1	0	2	1	100	0
51	0	4	0	1	0	0	0	0	3	0	180	0
52	0	3	0	0	0	0	0	0	3	0	160	0
53	0	5	0	0	0	0	0	0	3	0	93	0
54	0	0	0	0	0	1	0	0	3	0	87	0
55	1	7	0	0	0	0	0	0	1	0	166	0
56	1	4	1	0	0	0	0	1	3	0	168	0
57	0	6	0	0	0	1	1	0	4	1	145	0
58	0	8	0	0	0	0	0	0	3	0	207	1

APPENDIX A - Complete data set

ID	Cover letter	Pages CV	Photo	References	Vocabuary density	Words/ Page	Level of studies	Current field of studies	Education outside Sweden	Scholarship	Student associations (years)	Teaching/ lab assistant
59	0	1	0	0	0.86	100.00	1	O	0	0	0	0
60	0	2	1	0	0.54	382.00	0	O	0	0	0	0
61	0	1	1	0	0.64	218.00	0	O	0	0	1	1
62	0	2	1	0	0.73	77.00	1	CS	1	1	0	0
63	0	3	0	0	0.64	144.33	1	CS	1	1	0	0
64	1	1	0	0	0.49	411.00	1	SE	1	0	1	1
65	1	1	1	0	0.83	163.00	0	SE	0	1	0	0
66	0	3	1	0	0.52	263.00	0	CS	0	0	1	0
67	0	1	0	0	0.71	163.00	0	SE	0	0	2	0
68	0	2	0	0	0.64	187.00	1	CS	0	0	5	0
69	0	2	0	0	0.50	331.00	0	CS	0	0	0	1
70	1	2	0	1	0.66	186.50	1	CS	1	1	0	0
71	0	3	0	0	0.62	136.00	1	CS	1	0	0	0
72	0	3	0	0	0.55	243.00	1	CS	1	0	1	1

APPENDIX A - Complete data set

ID	GitHub/ BitBucket	Programming languages	Projects	Experience as developer	Own company	Android	Algorithms	Gender	Languages	Selected	Test score	Test > 200
59	0	5	0	1	0	0	1	0	3	1	205	1
60	1	4	1	1	0	1	0	0	2	0	210	1
61	0	4	0	0	0	0	0	1	1	0	120	0
62	0	12	1	0	0	0	1	0	4	1	87	0
63	0	9	1	0	0	1	1	0	3	1	201	1
64	0	8	0	0	0	1	0	0	2	0	280	1
65	0	6	0	1	1	0	0	0	1	0	97	0
66	0	9	0	1	0	1	1	0	3	1	62	0
67	0	4	0	1	0	1	0	1	2	0	153	0
68	0	4	0	0	0	0	1	0	1	1	252	1
69	1	6	1	1	0	1	1	0	1	1	136	0
70	1	4	0	1	0	0	0	0	2	0	191	0
71	0	10	1	1	0	1	1	0	3	1	50	0
72	1	7	1	1	0	0	1	0	3	1	241	1